Multimedia Information Retrieval at the Center for Intelligent Information Retrieval

R. Manmatha *
Multimedia Indexing and Retrieval Group
Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts, Amherst, MA 01003
manmatha@cs.umass.edu

Abstract

Abstract: Building the digital libraries of the future will require a number of different component technologies including the ability to retrieve multi-media information. This paper will describe progress in this area at the Center for Intelligent Information Retrieval (CIIR). This includes:

1. *Multi-modal retrieval using appearance based image retrieval and text retrieval. This work has been applied to a large database of trademarks containing image and text data from the US Patent and Trademark Office. 68,000 trademarks may be searched using either image retrieval or image and text retrieval while 615,000 trademarks may be searched using text retrieval.*

2. *Indexing handwritten manuscripts. Recently we have developed a scale-space technique for word segmentation in handwritten manuscripts.*

3. *Other projects including color based image retrieval and the extraction of text from images.*

1 Introduction

The Center for Intelligent Information Retrieval (CIIR) has a number of projects to index and retrieve multimedia information. We will describe some of the progress made in these areas since the last SDIUT meeting [20]. The projects include:

1. Image Retrieval: Work on indexing images using their content continues using both appearance based and color based retrieval. In previous work on appearance based retrieval [20, 26] we focussed on part image retrieval - whether a part of a query image is similar to a part of a database image. Recent work at the Center has focussed on "whole image retrieval' ' i.e. whether two images are similar in their entirety. In this work [27], two images are considered similar if their distributions of local curvature and phase at multiple scales are similar.

   The Center is also doing work on retrieving images, using color, from homogeneous databases. [6]. Color retrieval ssytems are inappropriate for heterogeneous databases. For example, a query image of a red flower will retrieve not only red flowers but also other red objects like cars and dresses. Most users do not find this meaningful. If, however, the database consisted only of flowers then a query on the color red would only retrieve red flowers and this is more meaningful to most users. This work has been applied to indexing a small database of flower patents. Another salient feature of this work is that using domain constraints, we are able to segment flowers from the background and index only the color of the flower rather than all the colors of the entire image.

2. Multi-modal retrieval: A multi-modal retrieval combining appearance based image retrieval and text retrieval is being applied to retrieve trademark images from a database provided by the US Patent and Trademark Office. 68000 trademarks may be searched using either image retrieval or image and text retrieval while 615,000 trademarks may be searched using text retrieval. A multi-modal provides many constraints so that the image search may be constrained. In addition, our multi-modal system solves the problem of how a query image is to be obtained. An initial search is done using text and the result is a list of trademarks with their associated text and images which may then be used for image or text retrieval.

This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademarks Office and the Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235, in part by NSF IRI-9619117 and in part by NSF Multimedia CDA-9502639. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

3. Finding Text in Images: The conversion of scanned documents into ASCII so that they can be indexed using INQUERY (CIIR's text retrieval engine). Current Optical Character Recognition Technology (OCR) can convert scanned text to ASCII but is limited to good clean machine printed fonts against clean backgrounds. Handwritten text, text printed against shaded or textured backgrounds and text embedded in images cannot be recognized well (if it can be recognized at all) with existing OCR technology. Many financial documents, for example, print text against shaded backgrounds to prevent copying.

   The Center has developed techniques to detect text in images. The detected text is then cleaned up and binarized and run through a commercial OCR. Such techniques can be applied to zoning text found against general backgrounds as well as for indexing and retrieving images using the associated text.

   The Center is continuing work in this area. Most of this work has involved speeding up some of our techniques in this area

4. Word Spotting: The indexing of hand-written and poorly printed documents using image matching techniques. Libraries hold vast collections of original handwritten manuscripts, many of which have never been published. Word Spotting can be used to create indices for such handwritten manuscript archives.

   Our recent work in this area has involved developing new techniques for word segmentation based on scale space methods. Old handwritten manuscripts are challenging for word segmentation algorithms for many reasons; ascenders and descenders from adjacent lines touch, noise and ink bleeding are present, the manuscripts show shine through and xeroxing and scanning have introduced additional artifacts. Our technique for word segmentation first involves segmenting the lines out using a new projection profile technique and then detecting words in each line by creating scale space blobs.

A discussion of the Center's work on word segmentation of handwritten mansucripts and its work on appearance based image retrieval and multi-modal retrieval now follows.

2 Appearance Based Image Retrieval and Multi-Modal Retrieval

The image intensity surface is robustly characterized using features obtained from responses to multi-scale Gaussian derivative filters. Koenderink [16] and others [11] have argued that the local structure of an image can be represented by the outputs of a set of Gaussian derivative filters applied to an image. That is, images are filtered with Gaussian derivatives at several scales and the resulting response vector locally describes the structure of the intensity surface. By computing features derived from the local response vector and accumulating them over the image, robust representations appropriate to querying images as a whole (global similarity) can be generated. One such representation uses histograms of features derived from the multi-scale Gaussian derivatives. Histograms form a global representation because they capture the distribution of local features (A histogram is one of the simplest ways of estimating a non parametric distribution). This global representation can be efficiently used for global similarity retrieval by appearance and retrieval is very fast.

The choice of features often determines how well the image retrieval system performs. Here the task is to robustly characterize the 3-dimensional intensity surface. A 3-dimensional surface is uniquely determined if the local curvatures everywhere are known. Thus, it is appropriate that one of the features be local curvature. The principal curvatures of the intensity surface are invariant to image plane rotations, monotonic intensity variations and further, their ratios are in principle insensitive to scale variations of the entire image. However, spatial orientation information is lost when constructing histograms of curvature (or ratios thereof) alone. Therefore we augment the local curvature with local phase, and the representation uses histograms of local curvature and phase.

Local principal curvatures and phase are computed at several scales from responses to multi-scale Gaussian derivative filters. Then histograms of the curvature ratios [15, 7] and phase are generated. Thus, the image is represented by a single vector (multi-scale histograms). During run-time the user presents an example image as a query and the query histograms are compared with the ones stored, and the images are then ranked and displayed in order to the user.

2.1 The choice of domain

There are two issues in building a content based image retrieval system. The first issue is technological, that is, the development of new techniques for searching images based on their content. The second issue is user or task related, in the sense of whether the system satisfies a user need. While a number of content based retrieval systems have been built ([10, 9]), it is unclear what the purpose of such systems is and whether people would actually search in the fashion described.

Here, we describe how the techniques described here may be scaled to retrieve images from a database of about 63000 trademark images provided by the US Patent and Trademark Office. This database consists of all (at the time the database was provided) the registered trademarks in the United States which consist only of designs (i.e. there are no words in them). Trademark images are

a good domain with which to test image retrieval. First, there is an existing user need: trademark examiners do have to check for trademark conflicts based on visual appearance. That is, at some stage they are required to look at the images and check whether the trademark is similar to an existing one. Second, trademark images may consist of simple geometric designs, pictures of animals or even complicated designs. Thus, they provide a test-bed for image retrieval algorithms. Third, there is text associated with every trademark and the associated text maybe used in a number of ways. One of the problems with many image retrieval systems is that it is unclear where the example or query image will come from. In this paper, the associated text is used to provide an example or query image. In future papers, we will explore how text and image searches may be combined to build more sophisticated systems. Using trademark images does have some limitations. First, we are restricted to binary images (albeit large ones). As shown later in the paper, this does not create any problems for the algorithms described here. Second, in some cases the use of abstract images makes the task more difficult. Others have attempted to get around it by restricting the trademark images to geometric designs [13].

2.2 Global representation of appearance

Three steps are involved in order to computing global similarity. First, local derivatives are computed at several scales. Second, derivative responses are combined to generate local features, namely, the principal curvatures and phase and, their histograms are generated. Third, the 1D curvature and phase histograms generated at several scales are matched. These steps are described next.

A. Computing local derivatives: Computing derivatives using finite differences does not guarantee stability of derivatives. In order to compute derivatives stably, the image must be regularized, or smoothed or band-limited. A Gaussian filtered image $I_\sigma = I * G$ obtained by convolving the image I with a normalized Gaussian $G(\mathrm{r}, \sigma)$ is a band-limited function. Its high frequency components are eliminated and derivatives will be stable. In fact, it has been argued by Koenderink and van Doorn [16] and others [11] that the local structure of an image I at a given scale can be represented by filtering it with Gaussian derivative filters (in the sense of a Taylor expansion), and they term it the N-jet.

However, the shape of the smoothed intensity surface depends on the scale at which it is observed. For example, at a small scale the texture of an ape's coat will be visible. At a large enough scale, the ape's coat will appear homogeneous. A description at just one scale is likely to give rise to many accidental mis-matches. Thus it is desirable to provide a description of the image over a number of scales, that is, a scale space description of the image. It has been shown by several authors [18, 14, 32, 30, 11], that under certain general constraints, the Gaussian filter forms a unique choice for generating scale-space. Thus local spatial derivatives are computed at several scales.

B. Feature Histograms: The normal and tangential curvatures of a 3-D surface (X,Y,Intensity) are defined as [11]:

$$N(\mathrm{p},\sigma) = \left[\frac{I_x^2 I_{yy} + I_y^2 I_{xx} - 2 I_x I_y I_{xy}}{\left(I_x^2 + I_y^2\right)^{\frac{3}{2}}}\right](\mathrm{p},\sigma)$$

$$T(\mathrm{p},\sigma) = \left[\frac{\left(I_x^2 - I_y^2\right) I_{xy} + \left(I_{xx} - I_{yy}\right) I_x I_y}{\left(I_x^2 + I_y^2\right)^{\frac{3}{2}}}\right](\mathrm{p},\sigma)$$

Where $I_x(\mathrm{p},\sigma)$ and $I_y(\mathrm{p},\sigma)$ are the local derivatives of Image I around point p using Gaussian derivative at scale σ. Similarly $I_{xx}(\cdot,\cdot)$, $I_{xy}(\cdot,\cdot)$, and $I_{yy}(\cdot,\cdot)$ are the corresponding second derivatives. The normal curvature N and tangential curvature T are then combined [15] to generate a shape index as follows:

$$C(\mathrm{p},\sigma) = atan\left[\frac{N+T}{N-T}\right](\mathrm{p},\sigma)$$

The index value C is $\frac{\pi}{2}$ when $N = T$ and is undefined when either N and T are both zero, and is, therefore, not computed. This is interesting because very flat portions of an image (or ones with constant ramp) are eliminated. For example in Figure 2(middle-row), the background in most of these face images does not contribute to the curvature histogram. The curvature index or shape index is rescaled and shifted to the range [0, 1] as is done in [7]. A histogram is then computed of the valid index values over an entire image.

The second feature used is phase. The phase is simply defined as $P(\mathrm{p},\sigma) = atan2(I_y(\mathrm{p},\sigma), I_x(\mathrm{p},\sigma))$. Note that P is defined only at those locations where C is and ignored elsewhere. As with the curvature index P is rescaled and shifted to lie between the interval [0, 1].

At different scales different local structures are observed and, therefore, multi-scale histograms are a more robust representation. Consequently, a feature vector is defined for an image I as the vector $V_i = \langle H_c(\sigma_1) \ldots H_c(\sigma_n), H_p(\sigma_1) \ldots H_p(\sigma_n)\rangle$ where H_p and H_c are the curvature and phase histograms respectively. We found that using 5 scales gives good results and the scales are $1 \cdots 4$ in steps of half an octave.

C. Matching feature histograms: Two feature vectors are compared using normalized cross-covariance defined as

$$d_{ij} = \frac{V_i^{(m)} \cdot V_j^{(m)}}{\left\|V_i^{(m)}\right\| \left\|V_j^{(m)}\right\|}$$

where $V_i^{(m)} = V_i - mean(V_i)$.

Retrieval is carried out as follows. A query image is selected and the query histogram vector V_q is correlated

with the database histogram vectors V_i using the above formula. Then the images are ranked by their correlation score and displayed to the user. In this implementation, and for evaluation purposes, the ranks are computed in advance, since every query image is also a database image.

2.2.1 Experiments

The curvature-phase method is tested using two databases. The first is a trademark database of 2048 images obtained from the US Patent and Trademark Office (PTO). The images obtained from the PTO are large, binary and are converted to gray-level and reduced for the experiments. The second database is a collection of 1561 assorted gray-level images. This database has digitized images of cars, steam locomotives, diesel locomotives, apes, faces, people embedded in different background(s) and a small number of other miscellaneous objects such as houses. These images were obtained from the Internet and the Corel photo-cd collection and were taken with several different cameras of unknown parameters, and under varying uncontrolled lighting and viewing geometry.

In the following experiments an image is selected and submitted as a query. The objective of this query is stated and the relevant images are decided in advance. Then the retrieval instances are gauged against the stated objective. In general, objectives of the form 'extract images similar in appearance to the query' will be posed to the retrieval algorithm. A measure of the performance of the retrieval engine can be obtained by examining the recall/precision table for several queries. Briefly, recall is the proportion of the relevant material actually retrieved and precision is the proportion of retrieved material that is relevant [31]. It is a standard widely used in the information retrieval community and is one that is adopted here.

Queries were submitted each to the trademark and assorted image collection for the purpose of computing recall/precision. The judgment of relevance is qualitative. For each query in both databases the relevant images were decided in advance. These were restricted to 48. The top 48 ranks were then examined to check the proportion of retrieved images that were relevant. All images not retrieved within 48 were assigned a rank equal to the size of the database. That is, they are not considered retrieved. These ranks were used to interpolate and extrapolate precision at all recall points. In the case of assorted images relevance is easier to determine and more similar for different people. However in the trademark case it can be quite difficult and therefore the recall-precision can be subject to some error. The recall/precision results are summarized in Table 1 and both databases are individually discussed below.

Figure 1 shows the performance of the algorithm on the trademark images. Each strip depicts the top 8 retrievals, given the leftmost as the query. Most of the shapes have roughly the same structure as the query. Note that, outline and solid figures are treated similarly (see rows one and two in Figure 1). Six queries were submitted for the purpose of computing recall-precision in Table 1.

Experiments are also carried out with assorted gray level images. Six queries submitted for recall-precision are shown in Figure 2. The left most image in each row is the query and is also the first retrieved. The rest from-left to right are seven retrievals depicted in rank order. Note that, flat portions of the background are never considered because the principal curvatures are very close to zero and therefore do not contribute to the final score. Thus, for example, the flat background in Figure 2(second row) is not used. Notice that visually similar images are retrieved even when there is some change in the background (row 1). This is because the dominant object contributes most to the histograms. In using a single scale poorer results are achieved and background affects the results more significantly.

The results of these examples are discussed below, with the precision over all recall points depicted in parentheses. For comparison the best text retrieval engines have an average precision of 50%:

1. Find similar cars(65%). Pictures of cars viewed from similar orientations appear in the top ranks because of the contribution of the phase histogram. This result also shows that some background variation can be tolerated. The eighth retrieval although a car is a mismatch and is not considered.

2. Find same face(87.4%) and find similar faces: In the face query the objective is to find the same face. In experiments with a University of Bern face database of 300 faces with a 10 relevant faces each, the average precision over all recall points for all 300 queries was 78%. It should be noted that the system presented here works well for faces with the same representation and parameters used for all the other databases. There is no specific "tuning" or learning involved to retrieve faces. The query "find similar faces" resulted in a 100% precision at 48 ranks because there are far more faces than 48. Therefore, it was not used in the final precision computation.

3. Find dark textured apes (64.2%). The ape query results in several other light textured apes and country scenes with similar texture. Although these are not mis-matches they are not consistent with the intent of the query which is to find dark textured apes.

4. Find other patas monkeys. (47.1%) Here there are 16 patas monkeys in all and 9 within a small view variation. However, here the whole image is being matched so the number of relevant patas monkeys is 16. The precision is low because the method cannot

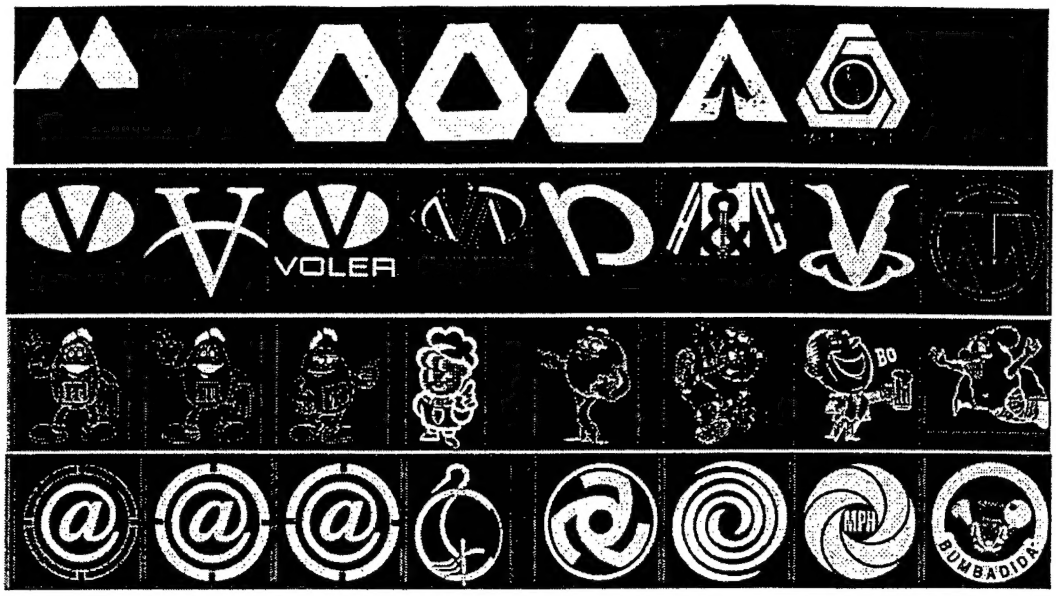

Figure 1: Trademark retrieval using Curvature and Phase

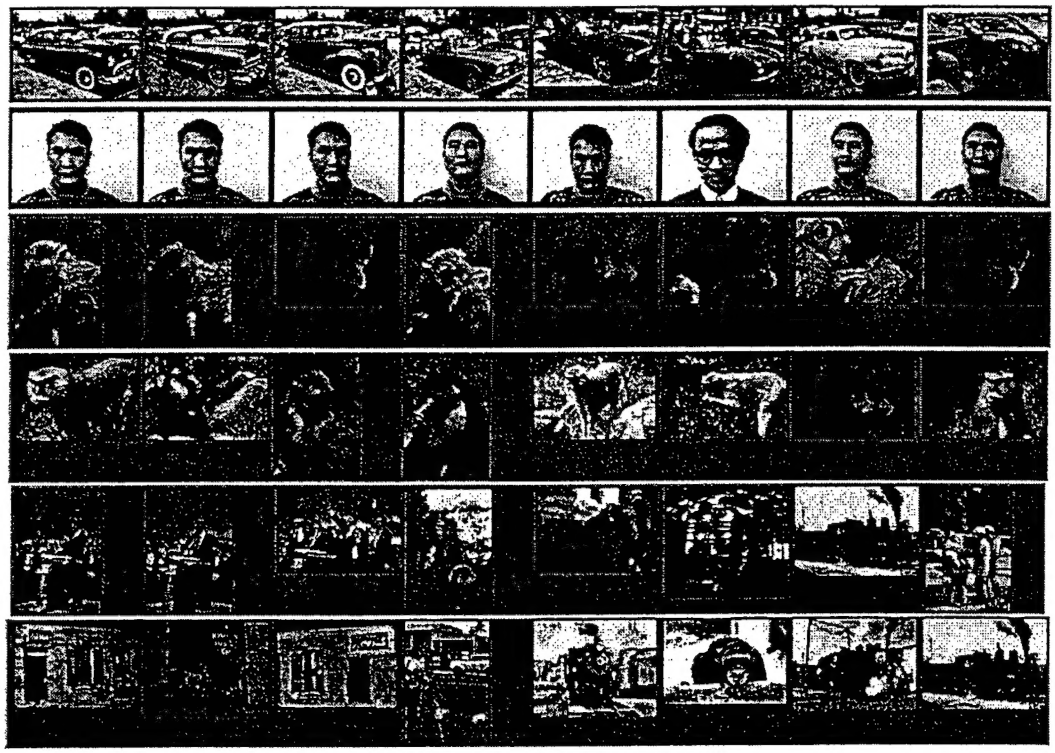

Figure 2: Image retrieval using Curvature and Phase

distinguish between light and dark textures, leading to irrelevant images. Note, that it finds other apes, dark textured ones, but those are deemed irrelevant with respect to the query.

5. Given a wall with a Coca Cola logo find other Coca Cola images (63.8%). This query clearly depicts the limitation of global matching. Although all three database images that had a certain texture of the wall (also had Coca Cola logos) were retrieved (100% precision), two other very dissimilar images with coca-cola logos were not.

6. Scenes with Bill Clinton (72.8%). The retrieval in this case results in several mismatches. However, three of the four are retrieved in succession at the top and the scenes appear visually similar.

While the queries presented here are not "optimal"

Table 1: Precision at standard recall points for six Queries

Recall	0	10	20	30	40	50	60	70	80	90	100
Precision(trademark) %	100	93.2	93.2	85.2	76.3	74.5	59.5	45.5	27.2	9.0	9.0
Precision(assorted) %	100	92.6	90.0	88.3	87.0	86.8	83.8	65.9	21.3	12.0	1.4

average(trademark)	61.1%
average(assorted)	66.3%

with respect to the design constraints of global similarity retrieval, they are however, realistic queries that can be posed to the system. Mismatches can and do occur. The first is the case where the global appearance is very different. The Coca Cola retrieval is a good example of this. Second, mismatches can occur at the algorithmic level. Histograms coarsely represent spatial information and therefore will admit images with non-trivial deformations. The recall/precision presented here compares well with text retrieval. The time per retrieval is of the order of milli-seconds. In the next section we discuss the application of the presented technique to a database of 63000 images.

2.3 Trademark Retrieval

The system indexes about 68,000 trademarks from the US Patent and Trademark office in the design only category. These trademarks are binary images. In addition, associated text consists of a design code that designates the type of trademark, the goods and services associated with the trademark, a serial number and a short descriptive text.

The system for browsing and retrieving trademarks is illustrated in Figure 3. The netscape/Java user interface has two search-able parts. On the left a panel is included to initiate search using text. Any or all of the fields can be used to enter a query. In this example, the text "Merriam Webster' is entered and all images associated with it are retrieved using the Inquery [4] text search engine. The user can then use any of the example pictures to search for images that are similar. In the specific example shown, The second image is selected and retrieved results are displayed on the right panel. The user can then continue to search using any of the displayed pictures as the query.

In this section we adapt the curvature/phase histograms to retrieve visually similar trademarks. The following steps are performed to retrieve images.

Preprocessing: Each binary image in the database is first size normalized, by clipping. Then they are converted to gray-scale and reduced in size.

Computation of Histograms: Each processed image is divided into four equal rectangular regions. This is different than constructing a histogram based on pixels of the entire image. This is because in scaling the images to a large collection, we found that the added degree of spatial resolution significantly improves the retrieval performance. The curvature and phase histograms are computed for each tile at three scale. A histogram descriptor of the image is obtained by concatenating all the individual histograms across scales and regions.

These two steps are conducted off-line.

Execution: The image search server begins by loading all the histograms into memory. Then it waits on a port for a query. A CGI client transmits the query to the server. Its histograms are matched with the ones in the database. The match scores are ranked and the top N requested retrievals are returned.

2.3.1 Examples

In Figure 3, the user typed in Merriam Webster in the text window. The system searches for trademarks which have either Merriam or Webster in th associated text and displays them. Here, the first two trademarks (first two images in the left window) belong to Merriam Webster. In this example, the user has chosen to 'click' the second image and search for images of similar trademarks. This search is based entirely on the image and the results are displayed in the right window in rank order. Retrieval takes a few seconds and is done by comparing histograms of all 63,718 trademarks on the fly.

The original image is returned as the first result (as it should be). The images in positions 2,3 and 5 in the second window all contain circles inside squares and this configuration is similar to that of the query. Most of the other images are of objects contained inside a roughly square box and this is reasonable considering that similarity is defined on the basis of the entire image rather than a part of the image.

The second example is shown in Figure 4. Here the user has typed in the word Apple. The system returns trademarks associated with the word Apple. The user queries using Apple computer's logo (the image in the second row, first column of the first window). Images retrieved in response to this query are shown in the right window. The first eight retrievals are all copies of Apple Computer's trademark (Apple used the same trademark for a number of other goods and so there are multiple copies of the trademark in the database). Trademarks number 9 and 10 look remarkably similar to Apple's trademark. They are considered valid trademarks because they are used for goods and services in areas other than computers. Trademark 13 is another version of Apple Computer's logo but with lines in the middle. Although somewhat visually different it is still retrieved

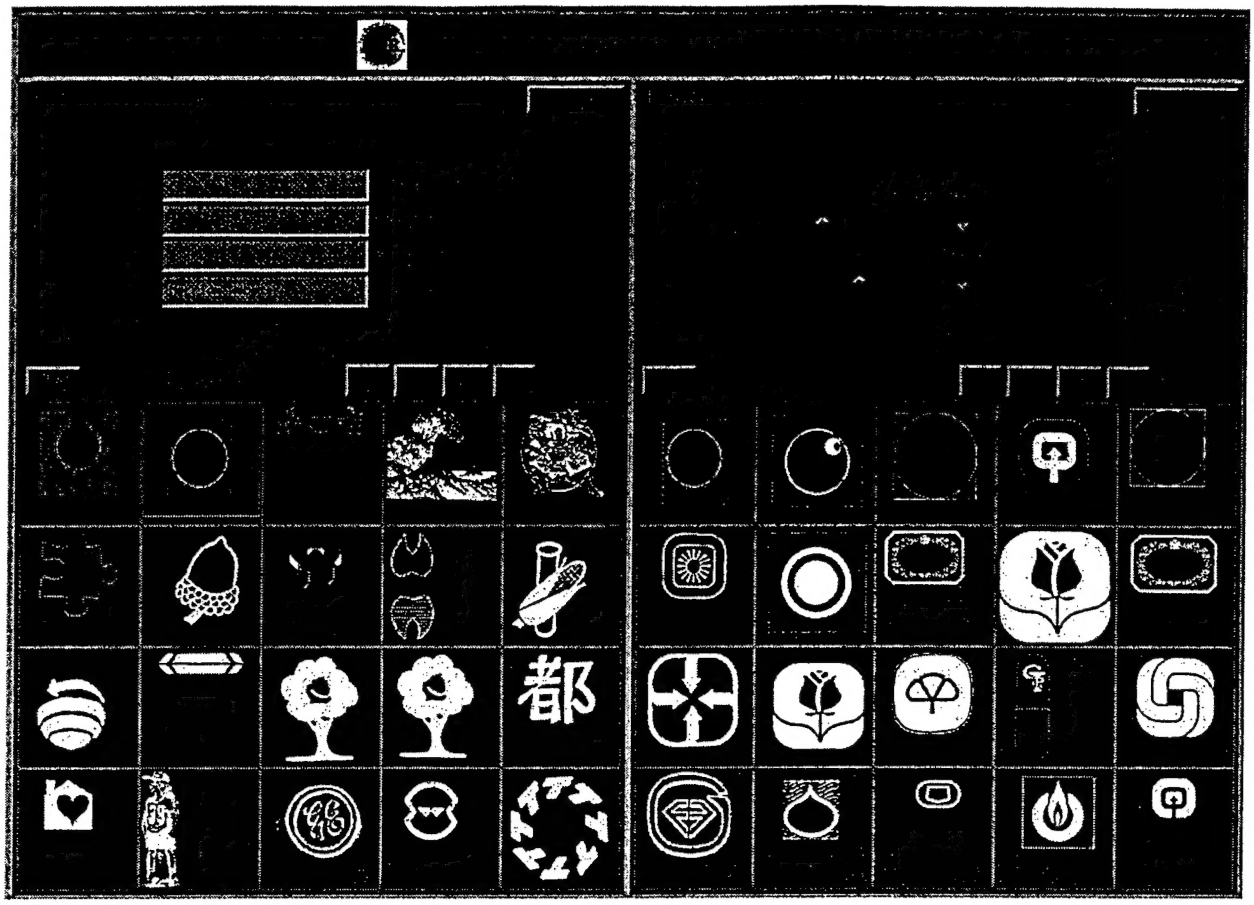

Figure 3: Retrieval in response to a "Merriam Webster" query

in the high ranks. Image 14 is an interesting example of a mistake made by the system. Although the image is not of an apple, the image has similar distributions of curvature and phase as is clear by looking at it.

The third example demonstrates combining text and visual appearance for searching. We use the same apple image obtained in the previous image as the image query. However, in the text box we now type "computer" and turn the text combination mode on. We now search for trademarks which are visually similar to the apple query image but also have the words computer associated with them. The results are shown in Figure 5 on the right-hand side. Notice that the first image is the same as the query image. The second image is an actual conflict. The image is a logo which belongs to the Atlanta Macintosh User's Group. The text describes the image as a peach but visually one can see how the two images may be confused with each other (which is the basis on which trademark conflicts are adjudicated). This example shows that it does not suffice to go by the text descriptions alone and image search is useful for trademarks. Notice that the fourth image which some people describe as an apple and others as a tomato is also described in the text as an apple.

The system has been tried on a variety of different examples of both two dimensional and three dimensional pictures of trademarks and had worked quite well. Clearly, there are issues of how quantitative results can be obtained for such large image databases (it is not feasible for a person to look at every image in the database to determine whether it is similar). In future work, we hope to evolve a mechanism for quantitative testing on such large databases. It will also be important to use more of the textual information to determine trademark conflicts.

3 Word Segmentation in Handwritten Archival Manuscripts

There are many single author historical handwritten manuscripts which would be useful to index and search. Examples of these large archives are the papers of George Washington, Margaret Sanger and W. E. B Dubois. Currently, much of this work is done manually. For example, 50,000 pages of Margaret Sanger's work were recently indexed and placed on a CDROM. A page by page index was created manually. It would be useful to automatically create an index for an historical archive similar to the index at the back of a printed book. To achieve this objective a semi-automatic scheme for indexing such documents have been proposed in [23, 22, 21]. In this scheme known as *Word Spotting* the document page is segmented into words. Lists

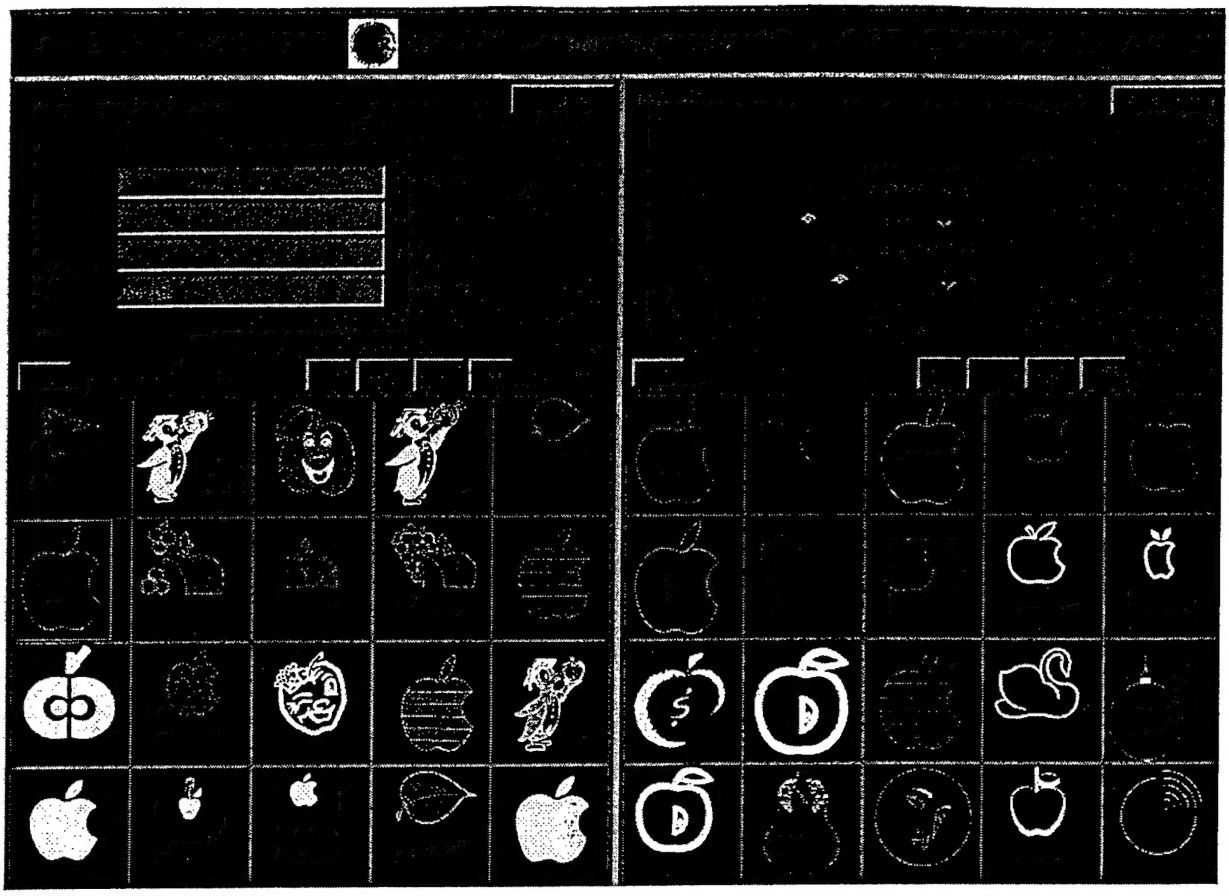

Figure 4: Retrieval in response to the query "Apple"

of words containing multiple instances of the same word are then created by matching word images against each other. A user then provides the ASCII equivalent to a representative word image from each list and the links to the original documents are automatically generated. The earlier work in [23, 22, 21] concentrated on the matching strategies and did not address full page segmentation issues in handwritten documents. In this paper, we propose a new algorithm for word segmentation in document images by considering the scale space behavior of blobs in line images.

Most existing document analysis systems have been developed for machine printed text. There has been little work on word segmentation for handwritten documents. Most of this work has been applied to special kinds of pages - for example, addresses or "clean" pages which have been written specifically for testing the document analysis systems. Historical manuscripts suffer from many problems including noise, shine through and other artifacts due to aging and degradation. No good techniques exist to segment words from such handwritten manuscripts. Further, scale space techniques have not been applied to this problem before.

We outline the various steps in the segmentation algorithm below.

The input to the system is a grey level document image. The image is processed to remove horizontal and vertical line segments likely to interfere with later operations. The page is then dissected into lines using projection analysis techniques modified for gray scale image. The projection function is smoothed with a Gaussian filter (low pass filtering) to eliminate false alarms and the positions of the local maxima (i.e., white space between the lines) is detected. Line segmentation, though not essential is useful in breaking up connected ascenders and descenders and also in deriving an automatic scale selection mechanism. The line images are smoothed and then convolved with second order anisotropic Gaussian derivative filters to create a scale space and the *blob* like features which arise from this representation give us the focus of attention regions (i.e., words in the original document image). The problem of automatic scale selection for filtering the document is also addressed. We have come up with an efficient heuristic for scale selection whereby the correct scale for blob extraction is obtained by finding the scale maxima of the blob extent. A connected component analysis of the blob image followed by a reverse mapping of the bounding boxes allows us to extract the words. The box is then extended vertically to include the ascenders and descenders. Our approach to word segmentation is novel as it is the first algorithm which utilizes the inherent scale space behavior of words

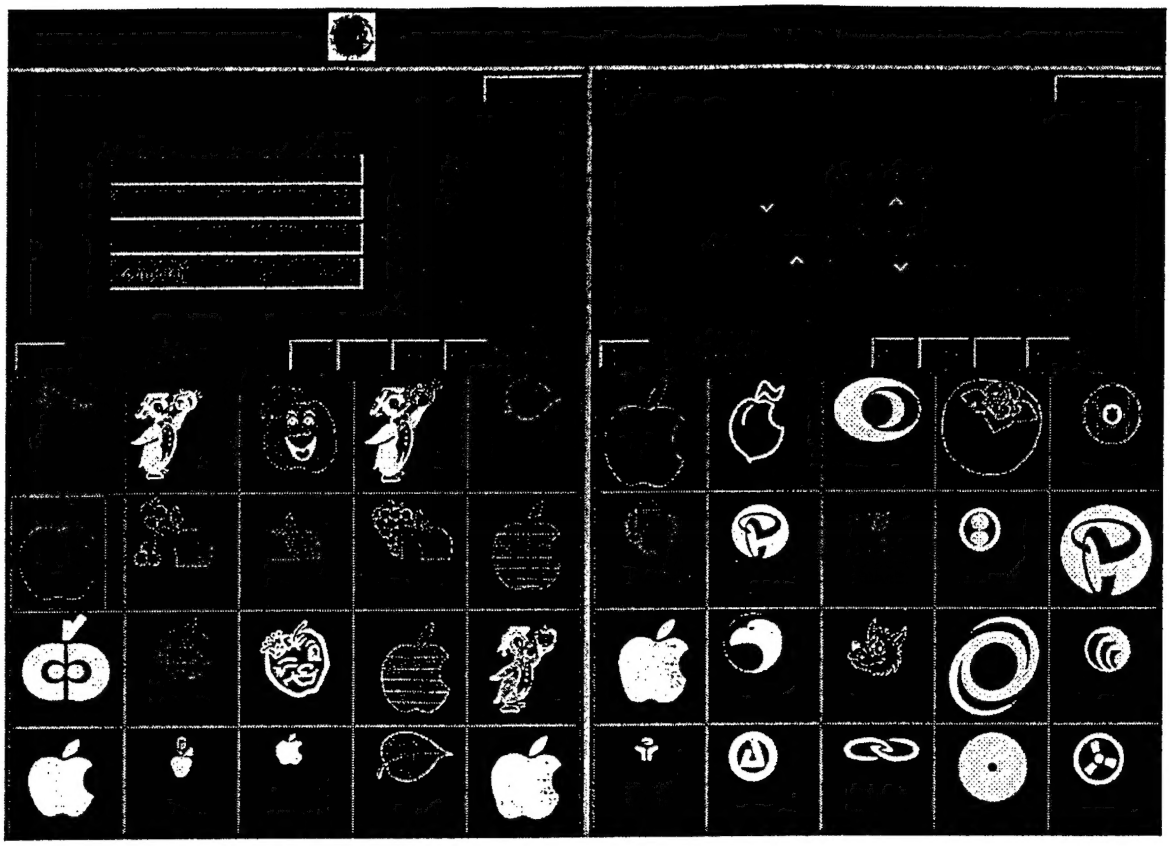

Figure 5: Retrieval in response to the query "Apple" limited to text searches

in grey level document images.

3.1 Related Work

Most recognition systems mask the issue of segmentation by considering well segmented patterns [2] or using words written in boxes whose location is known [8]. However, correct segmentation is crucial in full page document analysis and directly relates to the performance of the entire system. We present some of the work in word and character segmentation.

3.1.1 Word and character segmentation

Character segmentation schemes proposed in the literature have mostly been developed for machine printed characters and work poorly when extended to handwritten text. An excellent survey of the various schemes has been presented in [5].

Very few papers have dealt exclusively with issues of word segmentation in handwritten documents and even they have focussed on identifying gaps using geometric distance metrics between connected components. Seni and Cohen [28] evaluate eight different distance measures between pairs of connected component for word segmentation in handwritten text. In [19] the distance between the convex hulls is used. Srihari et all [29] present techniques for line separation and then word segmentation using a neural network. However, the existing word segmentation strategies have certain limitations

- Almost all the above methods require binary images. Also, they have been tried only on clean white self-written pages and not manuscripts.
- Most of the techniques have been developed for machine printed characters and not handwritten words. The difficulty faced in word segmentation is in combining discrete characters into words.
- Most researchers focus only on word recognition algorithms and considered a database of clean images with well segmented words, [1] is one such example.

Only a few [29] have performed full, handwritten page segmentation. However, we feel that schemes such as [29] are not applicable for page segmentation in manuscript images for the reasons mentioned below.

- Efficient image binarization is difficult on manuscript images containing noise and shine through.
- Connected ascenders and descenders have to be separated.
- Prior character segmentation was required to perform word segmentation and accurate character segmentation in cursive writing is a difficult problem. Also the examples shown are contrived (self written) and do not handle problems in naturally written documents.

3.2 Word Segmentation

Modeling the human cognitive processes to derive a computational methodology for handwritten word seg-

mentation with performance close to the human visual system is quite complex due to the following characteristics of handwritten text.

- The handwriting style may be cursive or discrete. In case of discrete handwriting characters have to be combined to form words.
- Unlike machine printed text, handwritten text is not uniformly spaced.
- Scale problem. For example, the size of characters in a header is generally larger than the average size of the characters in the body of the document.
- Ascenders and descenders are frequently connceted and words may be present at different orientations.
- Noise, artifacts, aging and other degradation of the document. Another problem is the presence of background handwriting or shine through.

We now present a brief background to scale space and how we have applied it to document analysis.

3.3 Scale space and document analysis

Scale space theory deals with the notion and importance of scale in any physical observation i.e. objects or features are relevant only at particular scales and meaningless at other scales [14, 11, 18]. In scale space, starting from an original image, successively smoothed images are generated along the scale dimension. It has been shown by several researchers [14, 11, 18] that the Gaussian uniquely generates the linear scale space of the image when certain conditions are imposed.

We feel that *scale space* also provides an ideal framework for document analysis. We may regard a document to be formed of features at multiple scales. Intuitively, at a finer scale we have characters and at larger scales we have words, phrases, lines and other structures. Hence, we may also say that there exists a scale at which we may derive words from a document image. We would, therefore, like to have an image representation which makes the features at that scale (words in this case) explicit : i.e. no further processing should be required to locate the words.

3.3.1 Formal definition

The linear scale space representation of a continuous signal with arbitrary dimensions consists of building a one parameter family of signals derived from the original one in which the details are progressively removed. Let $f\colon \Re^N \rightarrow \Re$ represent any given signal. Then, the scale space representation $I : \Re^N \times \Re_+ \rightarrow \Re$ is defined by letting the scale space representation at zero scale be equal to the original signal $I(\cdot; 0) = f$ and for $\sigma > 0$,

$$I(\cdot;\sigma) = G(\cdot;\sigma) \star f, \tag{1}$$

$$G(x,y;\sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{(2\sigma^2)}} \tag{2}$$

where G is the Gaussian kernel in two dimensions and σ is the scale parameter. We now describe the various stages in our algorithm.

3.4 Preprocessing

These handwritten manuscripts have been subjected to degradation such as fading and introduction of artifacts. The images provided to us are scanned versions of the photocopies of the original manuscripts. In the process of photocopying, horizontal and vertical black line segments/margins were introduced. Horizontal lines are also present within the text. The purpose of the preprocessing step is to remove some of these margins and lines so that they will not interfere with the blob analysis stage. The details of the pre-processing step are omitted here.

3.5 Line segmentation

Line segmentation allows the ascenders and descenders of consecutive lines to be separated. In the manuscripts it is observed that the lines consist of a series of horizontal components from left to right. Projection profile techniques have been widely used in line and word segmentation for machine printed documents [12]. In this technique a 1D function of the pixel values is obtained by projecting the binary image onto the horizontal or vertical axis. We use a modified version of the same algorithm extended to gray scale images. Let $f(x, y)$ be the intensity value of a pixel (x, y) in a gray scale image. Then, we define the vertical projection profile as

$$P(y) = \sum_{x=0}^{W} f(x,y) \tag{3}$$

where W is the width of the image. Figure 6 shows a section of an image (rotated by 90 deg.) in (a) and its projection profile in (b). The distinct local peaks in the profile corresponds to the white space between the lines and distinct local minima corresponds to the text (black ink). Line segmentation, therefore, involves detecting the position of the local maxima. However, the projection profile has a number of false local maxima and minima. The projection function $P(y)$ is therefore, smoothed with a Gaussian (low pass) filter to eliminate false alarms and reduce sensitivity to noise. A smoothed profile is shown in (c). The local maxima is then obtained from the first derivative of the projection function by solving for y such that :

$$P'(y) = P(y) \star G_y = 0 \tag{4}$$

The line segmentation technique is robust to variations in the size of the lines and has been tested on a wide range of handwritten pages. The next step after line segmentation is to create a scale space of the line images for blob analysis.

3.6 Blob analysis

Now we examine each line image individually to extract the words. A word image is composed of discrete char-

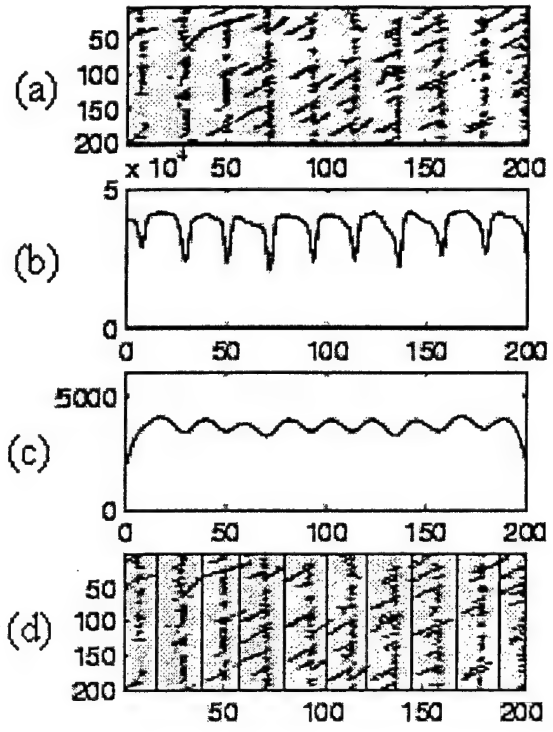

Figure 6: (a) A section of an image, (b) projection profile, (c) smoothed projection profile (d) line segmented image

acters, connected characters or a combination of the two. We would like to merge these sub-units into a single meaningful entity which is a word. This may be achieved by forming a blob-like representation of the image. A blob can be regarded as a connected region in space. The traditional way of forming a blob is to use a Laplacian of a Gaussian (LOG) [17] as the LOG is a popular operator and frequently used in blob detection and a variety of multi-scale image analysis tasks [3, 25, 17]. We have used a differential expression similar to a LOG for creating a multi-scale representation for blob detection. However, our differential expression differs in that we combine second order partial Gaussian derivatives along the two orientations at different scales. In the next section we present the motivation for using an anisotropic derivative operator.

3.6.1 Non uniform Gaussian filters

In this section some properties which characterize writing are used to formulate an approach to filtering words. In [17] Lindeberg observes that maxima in scale-space occur at a scale proportional to the spatial dimensions of the blob. If we observe a word we may see that the spatial extent of the word is determined by the following :

1. The individual characters determine the height (y dimension) of the word and
2. The length (x dimension) is determined by the number of characters in it.

A word generally contains more than one character and has an aspect ratio greater than one. As the x dimension of the word is larger than the y dimension, the spatial filtering frequency should also be higher in the y dimension as compared to the x dimension. This domain specific knowledge allows us to move from isotropic (same scale in both directions) to anisotropic operators. We choose the x dimension scale to be larger than the y dimension to correspond to the spatial structure of the word. Therefore, our approach for word segmentation, is based on the idea of a directional scale (i.e. generating an image representation by using Gaussian derivative operators at different scales for each of the two Cartesian coordinate axes) is in agreement with Lindeberg's observation that spatial dimensions are related to the scale. We define our anisotropic Gaussian filter as

$$G(x, y; \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2})} \quad (5)$$

We may also define the multiplication factor η as

$$\eta = \frac{\sigma_x}{\sigma_y} \quad (6)$$

In the scale selection section we will show that the average aspect ratio or the multiplication factor η lies between three and five for most of the handwritten documents available to us. Also the response of the anisotropic Gaussian filter (measured as the spatial extent of the *blobs* formed) is maximum in this range. For the above Gaussian, the second order anisotropic Gaussian differential operator $L_i(x, y; \sigma_x, \sigma_y)$ is defined as

$$L(x, y; \sigma_x, \sigma_y) = G_{xx}(x, y; \sigma_x) + G_{yy}(x, y; \sigma_y) \quad (7)$$

A scale space representation of the line images is constructed by convolving the image with equation 7 Consider a two dimensional image $f(x, y)$, then the corresponding output image is

$$\begin{aligned} I(x, y; \sigma_x, \sigma_y) &= G_{xx}(\cdot; \sigma_x) * f(x, y) \\ &\quad + G_{yy}(\cdot; \sigma_y) * f(x, y) \quad (8) \\ &= G_{xx}(\cdot; \sigma_x) * f(x, y) \\ &\quad + G_{yy}(\cdot; \eta\sigma_x) * f(x, y) \quad (9) \end{aligned}$$

The main features which arise from a scale space representation are blob-like (i.e., connected regions either brighter or darker than the background). The sign of I may then be used to make a classification of the 3-D intensity surface into foreground and background. For example consider the line image in Figure 7(a). The figures show the blob images $I(x, y; \sigma_x, \sigma_y)$ at increasing scale values. Figure 7(b) shows that at a lower scale the blob image consists of character blobs. As we increase

the scale, character blobs give rise to word blobs (Figure 7(c) and Figure 7(d)). This is indicative of the phenomenon of merging in blobs. It is seen that for certain scale values the blobs and hence the words are correctly delineated (Figure 7(d)). A further increase in the scale value may not necessarily cause word blobs to merge together and other phenomenon such as splitting is also observed. These figures show that their exists a scale at which it is possible to delineate words. In the next section we present an approach to automatic scale selection for blob extraction.

(a) A line image

(b) Blob image at scale $\sigma_y = 1, \sigma_x = 2$

(c) Blob image at scale $\sigma_y = 2, \sigma_x = 4$

(d) Blob image at scale $\sigma_y = 4, \sigma_x = 16$

(e) Blob image at scale $\sigma_y = 6, \sigma_x = 36$

Figure 7: A line image and the output at different scales

3.7 Choice of scale

Scale space analysis does not address the problem of scale selection. The solution to this problem depends on the particular application and requires the use of prior information to guide the scale selection procedure. Some of our work in scale selection draws motivation from Lindeberg's observation [17] that the maximum response in both scale and space is obtained at a scale proportional to the dimension of the object. A document image consists of structures such as characters, words and lines at different scales. However, as compared to other types of images, document images have this unique property that a large variation in scale is not required to extract a particular type of structure. For example, all the words are essentially close together in terms of their scale and therefore, can be extracted without a large variation in the scale parameter. Hence, there exists a scale where each of the individual word forms a distinct blob. The output (blob) is then maximum at this value of the scale parameter. We show elsewhere [24] that this scale is a function of the vertical dimension of the word if the aspect ratio is fixed.

Our algorithm requires selecting σ_y and the multiplication factor η for blob extraction.

A base scale is obtained by using the height of the line: i.e., an estimate of σ_y is obtained as a fraction of the line height.

$$\sigma_y = k \times \text{Line height} \tag{10}$$

where $0 < k < 1$, the nearby scales are then examined to determine the maximum over scales. For our specific implementation we have used $k = 0.1$ and sampled σ_y at intervals of 0.3. The two values were determined experimentally and worked well over a wide range of images. The scales are then picked.

The details of the scale selection process are given elsewhere [24]. Breifly, by plotting a graph which shows the extent of the blobs versus the *eta* for a constant σ_y, we have shown that the maximum usually occurs for values of η between 3 and 5 (see [24]) for a large number of images. Thus, we choose $\eta = 4$.

3.8 Blob extraction and post processing

After the word blobs have been obtained at the correct scale they define the focus of attention regions which correspond to the actual words. Hence, these blobs have to be mapped back to the original image to locate the words. A widely used procedure is to enclose the blob in a bounding box which can be obtained through connected component analysis. In a blob representation of the word, localization is not maintained. Also parts of the words, especially the ascenders and descenders, are lost due to the earlier operations of line segmentation and smoothing (blurring). Therefore, the above bounding box is extended in the vertical direction to include these ascenders and descenders. At this stage an area/ratio filter is used to remove small structures due to noise.

3.9 Results

The technique was tried on 30 randomly picked images from different sections of the George Washington corpus of 6,400 images and a few images from the archive of papers of Erasmus Hudson. This allowed us to test on algorithm on wide range of handwritten documents such as letters, notebook pages etc. To reduce the runtime, the images have been smoothed and sub-sampled to a quarter of their original size. The algorithm takes 120 seconds to segment a document page of size 800 x

600 pixels on a PC with a 200 MHz pentium processor running LINUX. A segmentation accuracy ranging from $77 - 96$ percent with an average accuracy around 87.6 percent was observed. Figure 8 shows a segmented image with bounding boxes drawn on the extracted words. The method worked well even on faded, noisy images and Table 1 shows the results averaged over a set of 30 images.

The first column indicates the average no. of distinct words in a page as seen by a human observer. The second column indicates the % of words detected by the algorithm i.e, words with a bounding box around them, this includes words correctly segmented, fragmented and combined together. This measure is required as some of the words may be sufficiently small or faint to be mistaken for noise or an artifact. The next column indicate the % of words fragmented. Word fragmentation occurs if a character or characters in a word have separate bounding boxes or if 50 percent or greater of a character in a word is not detected. Line fragmentation occurs due to the dissection of the image into lines. A word is line fragmented if 50 percent or greater of a character lies outside the top or bottom edges of the bounding box. The sixth column indicates the words which are combined together. These are multiple words in the same bounding box and occur due to the choice of a larger scale in segmentation. The last column gives the percentage of correctly segmented words.

Avg. words per image	% words detected	% fragmented words +line	% words combined	% words correctly segmented
220	99.12	1.75 +0.86	8.9	87.6

Table 2: Table of segmentation results

4 Conclusion

This paper has described the multimedia indexing and retrieval work being done at the Center for Intelligent Information Retrieval. We have described work on a system for multi-modal retrieval combining text and image retrieval as well as word segmentation for handwritten archives. The research described is part of an on-going research effort focused on indexing and retrieving multimedia information in as many ways as possible. The work described here has many applications, principally in the creation of the digital libraries of the future.

5 Acknowledgements

A number of students, staff and faculty have contributed to the work described in this paper. The appearance based image retrieval work was done by S. Chandu Ravela while the color based retrieval work was done by Madirakshi Das. Victor Wu worked on extracting text from images while Nitin Srimal worked on word segmentation using scale space blobs. Tom Michel and Kamal Souccar worked on the text retrieval as well the interfaces for the multi-modal retrieval. Joseph Daverin, David Hirvonen and Adam Jenkins provided programming support for different parts of this work. Bruce Croft contributed to the text and multi-modal retrieval and James Allan provided suggestions on text retrieval.

References

[1] A.J. Robinson A.W. Senior. An off-line cursive handwriting recognition system. *IEEE transactions on PAMI*, 3:309–321, 1998.

[2] H. S. Baird and K. Thompson. Reading Chess. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(6):552–559, 1990.

[3] D. Blostein and N. Ahuja. A multiscale region detector. *Computer Vision Graphics and Image Processing*, 45:22–41, 1989.

[4] J. P. Callan, W. B. Croft, and S. M. Harding. The inquery retrieval system. In *Proceedings of the 3rd International Conference on Database and Expert System Applications*, pages 78–83, 1992.

[5] R. G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transactions on PAMI*, 18:690–706, July 1996.

[6] M. Das, R. Manmatha, and E. M. Riseman. Indexing flowers by color names using domain knowledge-driven segmentation. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV'98)*, pages 94–99, Princeton, NJ, Oct. 1998.

[7] Chitra Dorai and Anil Jain. Cosmos - a representation scheme for free form surfaces. In *Proc. 5th Intl. Conf. on Computer Vision*, pages 1024–1029, 1995.

[8] R. O. Duda and P. E. Hart. Experiments in recognition of hand-printed text. In *AFIPS Conference Proceedings*, pages 1139–1149, 1968.

[9] J.R. Bach et al. The virage image search engine: An open framework for image management. In *SPIE conf. on Storage and Retrieval for Still Image and Video Databases IV*, pages 133–156, 1996.

[10] Myron Flickner et al. Query by image and video content: The qbic system. *IEEE Computer Magazine*, pages 23–30, Sept. 1995.

[11] L. Florack. *The Syntactical Structure of Scalar Images*. PhD thesis, University of Utrecht, Utrecht, Holland, 1993.

[12] J. Ha, R. M. Haralick, and I. T. Phillips. Document page decomposition by the bounding-box projection technique. In *ICDAR*, pages 1119–1122, 1995.

[13] K. Shields J. P. Eakins and J. M. Boardman. Artisan - a shape retrieval system based on boundary family indexing. In *In Proc. SPIE conf. on Storage and Retrieval for Image and Video Databases IV, vol. 2670, San Jose*, pages 17–28, Feb 1996.

[14] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–396, 1984.

[15] J. J. Koenderink and A. J. Van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8), 1992.

[16] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[17] T. Lindeberg. On scale selection for differential operators. In *Eighth Scandinavian Conference on Image Analysis*, pages 857–866, 1993.

[18] Tony Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.

[19] U. Mahadevan and R. C. Nagabushnam. Gap metrics for word separation in handwritten lines. In *ICDAR*, pages 124–127, 1995.

[20] R. Manmatha. Multimedia indexing and retrieval at the center for intelligent information retrieval. In *Symposium on Document Image Understanting Technology, SDIUT'97*, Cambridge, U.K., April 1997. 4th European Conf. Computer Vision, Institute for Advanced Computer Studies, University of Maryland.

[21] R. Manmatha and W. B. Croft. Word spotting: Indexing handwritten manuscripts. In Mark Maybury, editor, *Intelligent Multi-media Information Retrieval*. AAAI/MIT Press, April 1998.

[22] R. Manmatha, Chengfeng Han, and E. M. Riseman. Word spotting: A new approach to indexing handwriting. In *Proc. Computer Vision and Pattern Recognition Conference*, pages 631–637, 1996.

[23] R. Manmatha, Chengfeng Han, E. M. Riseman, and W. B. Croft. Indexing handwriting using word matching. In *Digital Libraries '96: 1st ACM International Conference on Digital Libraries*, pages 151–159, 1996.

[24] R. Manmatha and N. Srimal. Scale space technique for word segmentation in handwritten manuscripts. In *submitted to the IEEE International Conference on Computer Vision (ICCV'99)*, Sep. 1999.

[25] D. Marr. *Vision*. W.H. Freeman: San Francisco, 1982.

[26] S. Ravela and R. Manmatha. Image retrieval by appearance. In *In the Proc. of the 20th Intl. Conf. on Research and Development in Information Retrieval (SIGIR'97)*, pages 278–285, July 1997.

[27] S. Ravela and R. Manmatha. On computing global similarity in images. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV'98)*, pages 82–87, Princeton, NJ, Oct. 1998.

[28] G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27:41–52, 1994.

[29] S. Srihari and G. Kim. Penman : A system for reading unconstrained handwritten page images. In *Symposium on document image understanding technology (SDIUT 97)*, pages 142–153, April 1997.

[30] Bart M. ter Har Romeny. *Geometry Driven Diffusion in Computer Vision*. Kluwer Academic Publishers, 1994.

[31] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[32] A. P. Witkin. Scale-space filtering. In *Proc. Intl. Joint Conf. Art. Intell.*, pages 1019–1023, 1983.

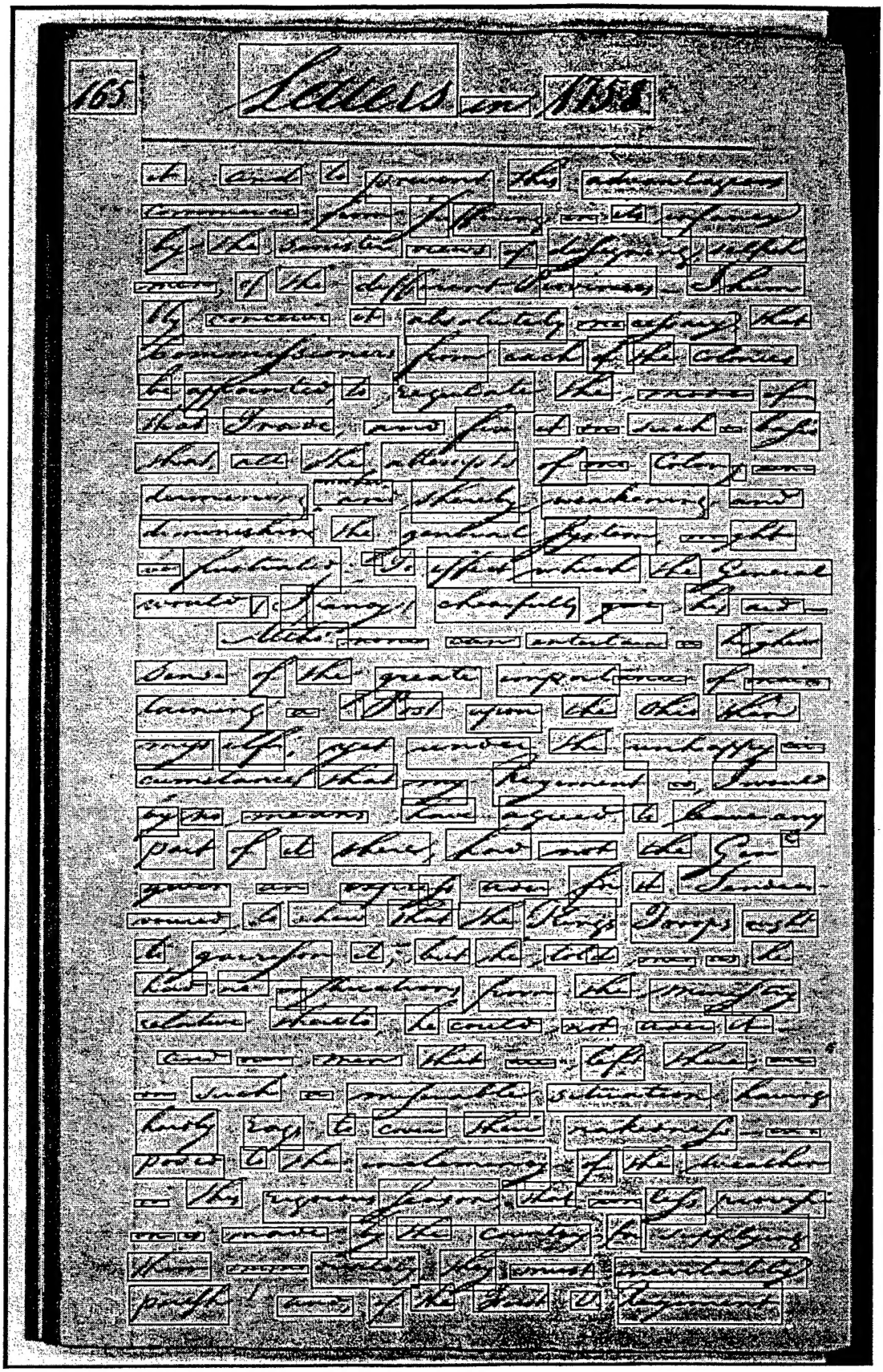

Figure 8: Segmentation result on a image 1670165.tif from the George Washington collection